Videopuzzle: Descriptive one-shot video composition

VideoPuzzle: Descriptive One-Shot Video Composition

Qiang Chen, Meng Wang, Zhongyang Huang, Yang Hua, Zheng Song, and Shuicheng Yan

***Abstract*—A large amount of short, single-shot videos are created by personal camcorder every day, such as the small video clips in family albums, and thus a solution for presenting and managing these video clips is highly desired. From the perspective of professionalism and artistry, long-take/shot video, also termed one-shot video, is able to present events, persons or scenic spots in an informative manner. This paper presents a novel video composition system "Video Puzzle" which generates aesthetically enhanced long-shot videos from short video clips. Our task here is to automatically composite several related single shots into a virtual long-take video with spatial and temporal consistency.**

We propose a novel framework to compose descriptive long-take video with content-consistent shots retrieved from a video pool. For each video, frame-by-frame search is performed over the entire pool to find start-end content correspondences through a coarse-to-fine partial matching process. The content correspondence here is general and can refer to the matched regions or objects, such as human body and face. The content consistency of these correspondences enables us to design several shot transition schemes to seamlessly stitch one shot to another in a spatially and temporally consistent manner. The entire long-take video thus comprises several single shots with consistent contents and fluent transitions. Meanwhile, with the generated matching graph of videos, the proposed system can also provide an efficient video browsing mode. Experiments are conducted on multiple video albums and the results demonstrate the effectiveness and the usefulness of the proposed scheme.

***Index Terms*—Image retrieval, one-shot video, video authoring, video transition.**

I. Introduction

With the popularity of personal digital devices, the amount of home video data is growing explosively. These digital videos have several characteristics: (1) compared with former videos recorded by non-digital camcorder, nowadays videos are usually captured more casually due to the less constraint of storage, and thus the number of clips is often quite large; (2) many videos may only contain a single shot and are very short; and (3) their contents are diverse yet related with few major subjects or events. Users often need to maintain their own video clip collections captured at different locations and time. These unedited and unorganized videos bring difficulties to their management and manipulation. For example, when users want to share their story with others over video sharing websites and social networks, such as YouTube.com and Facebook.com, they will need to put more efforts in finding, organizing and uploading the small video clips. This could be an extremely difficult **"Puzzle"** for users. Previous efforts towards efficient browsing such large amount of videos mainly focus on video summarization. These methods aim to capture the main idea of the video collection in a broad way, which, however, are not sufficiently applicable for video browsing and presentation. In this paper, we further investigate how to compose a content-consistent video from a video collection with an aesthetically attractive one-shot presentation. One-shot videos or long-shot video,[1] also known as long-take video (we will exchangeably use them hereafter), means a single shot that is with relatively long duration. Long shot has been widely used in the professional film industry, MTV video[2] and many other specific video domains owing to its uniqueness in presenting comprehensive content in a continuous and consistent way. However, capturing a high-quality long-shot video needs an accurate coordination between the camera movement and the captured object for a long period, which is usually difficult even for professionals.

In this paper, we introduce a scheme, "Video Puzzle", which can automatically generate a virtual one-shot presentation from multiple video clips. Given a messy collection of video clips, Video Puzzle can select a clip subset with consistent major topic (similar with finding the clues and solving the Puzzle Games among the images [16]). The topic can refer to a person, object, or a scene here. It can be specified by users or found with an automatic discovery method. The start-end frame correspondences of these clips are then established with an efficient coarse-to-fine method, and we compose them into a long clip in a seamless manner accordingly, i.e., a one-shot presentation. Therefore, Video Puzzle provides a novel presentation of video content that enables users to have a deeper impression of the

Manuscript received January 06, 2012; revised May 04, 2012; accepted September 08, 2012. Date of publication December 24, 2012; date of current version March 13, 2013. This work was supported in part by the National Basic Research Program of China (973 Program) (Grant No. 2013CB329604) and the Natural Science Foundation of China (Grant No: 61272393). This work was also supported by Singapore Ministry of Education under research Grant MOE2010-T2-1-087. The work was performed when Q. Chen did his part-time industry intern in Panasonic Singapore Laboratories. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sheng-Wei (Kuan-Ta) Chen.

Q. Chen, Z. Song, and S. Yan are with the Department of Electrical and Computer Engineering, National University of Singapore (e-mail: chenqiang@nus.edu.sg; zheng.s@nus.edu.sg; eleyans@nus.edu.sg).

M. Wang is with the Hefei University of Technology, China (e-mail: eric.mengwang@gmail.com).

Z. Huang and Y. Hua are with the Panasonic Singapore Laboratories, Panasonic R&D Centre Singapore (e-mail: zhongyang.huangsg.panasonic.com; yang.huasg.panasonic.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2012.2236306

[1]Sometimes "long shot" is also used for indicating shot size, i.e., the distance between camera and the captured object. Here we emphasize that the "long" indicates duration in our work.

[2]This one-shot MTV has been watched 58,213,601 times on Youtube at Apr 23, 20:00PM, PST, 2011. http://www.youtube.com/watch?v=m-jRHI3INyg

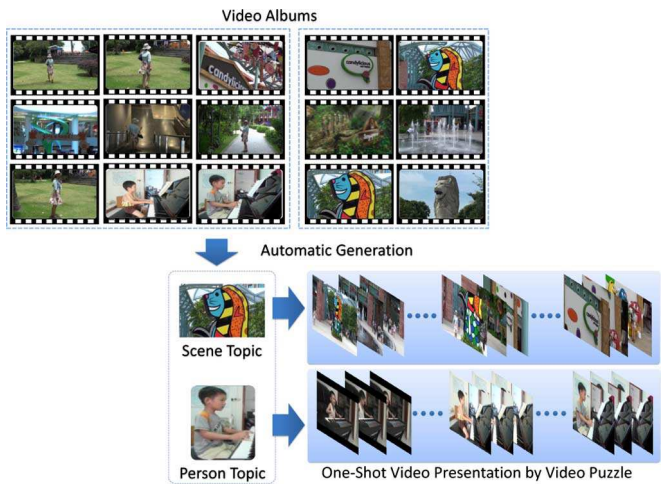


Fig. 1. An illustration of the Video Puzzle presentation scheme, which generates one-shot videos by selecting and composing short video clips.

story within the video collection. Fig. 1 shows the working process of Video Puzzle via two examples. The system can automatically discover video clips with "similar/continuous topics" in a video album and naturally stitch them into a single virtual long-take video, which can yield a cohesive presentation and convey a consistent underlying story. It is challenging as 1) it is generally hard to find shots which can be naturally combined among a large amount of candidate videos, and 2) generating seamless transition between video shots is difficult usually.

The contribution of our work can be summarized as follows:

(1) We propose a video puzzle scheme. It is able to extract video contents about a specific topic and compose them into a virtual one-shot presentation. The scheme is flexible and several components can be customized and applied to different applications.

(2) We propose an efficient method to find the content correspondences of multiple videos and then compose them into a clip with an optimized approach.

(3) We introduce two applications based on the video puzzle scheme, one about home video presentation and the other about landmark video generation.

Specifically, the two specific applications introduced are:

(1) Personal video presentation. With a large set of personal video contents, we can generate a video matching graph which explicitly shows the content-consecutive relation of videos. The storyline of the video album found by Video Puzzle will automatically pop up. Besides, user only needs to appoint a specific person or scene and then we can generate a one-shot presentation to describe the corresponding person or scene by mining the video graph.

(2) Comprehensive landmark video generation. With multiple web videos that describe the same landmark, we are able to generate a one-shot visual description of the landmark, which contains more comprehensive visual description of the landmark, such as the visual contents captured from different views.

In comparison with the conventional video abstraction and presentation techniques, we not only provide a novel presentation approach (a virtual one-shot video) but also facilitate further services such as editing.

The rest of this paper is organized as follows. Section II reviews related work on video presentation and video editing. In Section III, we give a system overview of our solution. We present the implementation details in terms of partial video matching in Section IV. Section V formulates the one-shot video generation process as a path finding problem. The transition generation is introduced in Section VI. Evaluation is provided in Section VII, followed with concluding remarks in Section VIII.

II. Related Work

One preliminary work that is worth mentioning is [2]. It is the first work that proposes to compose coherent presentation automatically if there are appropriate domain-specific metadata associated with video segments and the composition techniques are established. Another preliminary work is [17]. The system automatically selects home video segments and aligns them with music to create an edited video segment which is quite different from ours. Our system concentrates on how to provide consecutive smooth video while theirs try to fit the video segment with the music. Other approaches [37] also endeavor to classify the segments by film theory, and compose them into a story. However, the target of the method is for professional videos which capture the whole story of a certain event while home videos and web videos often have no fixed single story.

A. Video Summarization

Many previous works focus on producing effective video summarization with visual friendliness and in a compact form. Existing methods can be classified into two categories, i.e., dynamic representation and static representation. Dynamic representation generates a video sequence that is composed of a series of sub-clips extracted from one or multiple video sequences or generated from a collection of photos [11], [25], [31], whereas static representation generally generates one or multiple images from video key-frames to facilitate not only their viewing but also transmission and storage [4], [6], [8], [20], [29], [36]. Although video summarization can reduce the cost for video browsing, there is a risk of missing details and the possibly inaccurate summarization also may cause inconvenience in browsing.

For static representation, Uchihashi *et al.* [36] propose a video management system for generating story board and Calic *et al.* [6] propose a comic-like video summarization algorithm. Chiu *et al.* [8] provide a solution targeting for small display on mobile devices. A morphological grouping technique is described for finding 30 regions of high activity or motion from a video embedded in an image plane. The representation of motion in static images is a complex task with roots in art and science [13]. Caspi *et al.* [7] also propose a video summarization system which reduces browsing time, minimizes screen-space utilization, while preserving the crux of the video content and the sensation of motion. Mei *et al.* [29] present an automatic procedure for constructing a compactly synthesized

image collage from a video sequence. Boreczky [47] propose to select still images from a video suitable for summarizing the video and for providing entry points into it. These approaches are intended to represent the story line in a image/video, but they do not satisfy certain desired properties of visual representation, such as coherence and continuity.

For dynamic representation, Lee *et al.* [25] provide a scenario-based dynamic video abstractions using graph matching. Scharcanski *et al.* [31] propose a hierarchical technique to identify clinically relevant segments in diagnostic hysteroscopy videos and their associated key-frames, and then create a rich video summary. This approach is adaptive to video contents, and it represents the clinically relevant video segments hierarchically to facilitate fast video browsing. Correa *et al.* [11] present a system for generating dynamic narratives from videos. These narratives are characterized for being compact, coherent and interactive. This system can be used to create interactive posters for video clips. Barnes *et al.* [3] propose a multiscale tapestry which represents an input video as a seamless and zoomable summary image which can be used to navigate through the video.

B. Video Editing/Composition

Our work is also related to video editing and composition. In comparison with still image editing, content-based video editing faces the additional challenges of maintaining the spatial-temporal consistency with respect to geometry. This brings up difficulties of seamlessly modifying video contents, such as inserting or removing an object. Zhang *et al.* [44] provide a solution based on an unsupervised inference of view-dependent depth maps for all video frames. Yan *et al.* [43] transfer desired features from a source video to the target video such as colorizing videos, reducing video blurs, and video rhythm adjustment. Recently, Wang *et al.* [38] have studied automatic broadcast soccer video composition. There also exist studies on video texture [5], [23], [32] which aims to provide a continuous and infinitely varying stream of images. Rav-Acha *et al.* [30] explored time flow manipulation in video, such as the creation of new videos in which events that occurred at different times are displayed simultaneously. Our proposed seamless video composition technique is inspired from these works and we also integrate the object-level matching into the video composition procedure. Although our work and [40], [41] can provide aesthetically pleasing form videos among the user's video collection, the targets and methodologies used are totally different. Our system aims to automatically discover content-consistent video shots and compose into a virtual long-take video with spatial and temporal consistency while [40] aims to provide a tree structure collection with temporal smoothing for ease of video browsing. Besides, our system displays the collection with a more general graph structure and inferencing on the graph leads to the auto-discover of content-consistent video shots. Moreover, the video similarity in our system is based on mutli-cue matching and no previous similar work on video browsing has used this kind of information as far as we know. Video Textures [32] is a kind of temporal composition techniques. But it is within the scope of a single video and based upon visual similarity only. Our composition focuses on between-video transition with partial-matched video content. It is worth noting that Kobayashi et.al [45] extracts a video object from each video frame and creates locally consistent video sequences using a 2D motion graph. However, it can only deal with static background for foreground extraction. Kemelmacher *et al.* [46] propose a new photo exploration way that generates face animations from large image collections of the same person. It is close to our proposed object-oriented matching transition, but our method works more generally and focuses on video. There are also some works about motion graphs [22] which use parameterized motion capture data to construct smooth transition. However, we are dealing with real practical video data and these works are not applicable here. We need to mention that there exist several media composition works that all extract a media subset through finding a path in a graph constructed by media samples, such as [16], [22] [40]. But a major contribution of our work is the coarse-to-fine process for identifying the correspondence of video clips (i.e., the media graph construction process)

The overall scheme "Video Puzzle" aims to discover content-consistent video shots and composes them into a virtual long-take video. To this end, we propose a novel graph-based visualization and path finding approach. The graph is constructed based on geometry matching (homograph mapping) and object matching (human, face). Based on the multi-cue content matching, the transition of video shots becomes meaningful and seamless.

III. An Overview of the Scheme

Our task is to automatically compose several related video shots into a virtual long-take video with spatial and temporal consistency, and it is different from the traditional works that try to either find a group of similar video clips or fit the composed video with extra information such as music or metadata [2], [17]. For a given video collection that contains N video clips $\{V_i, i = 1, \ldots, N\}$,[3] the system mainly contains three key components, as illustrated in Fig. 2.

Firstly, we implement a coarse-to-fine partial matching scheme to generate a matching graph of the video collection. The matching scheme serves as a three-level matching, i.e., video pair selection, sequence-sequence correspondence finding, and frame-level exact matching. The video pair selection acts as an evidence for ensuring the *non-redundant* and *complete* quality of the generated one-shot video. It uses a hashing-based method to quickly obtain the video similarity measurement. We then find sequence correspondence of the selected video pairs through local keypoints matching. The final frame-level matching aims to find different matched objects to provide variant and rich clues for video transition generation. We implement three object matching methods in this part, i.e., salient object matching using local visual pattern discovery [26], and human and face appearance matching based on automatic human and face localization [15], [18].

[3]For the sake of simplicity, we assume that each video clip contains only a single shot. For multiple-shot videos, we can easily split them into single shots by the state-of-the-art video shot detection techniques [12].

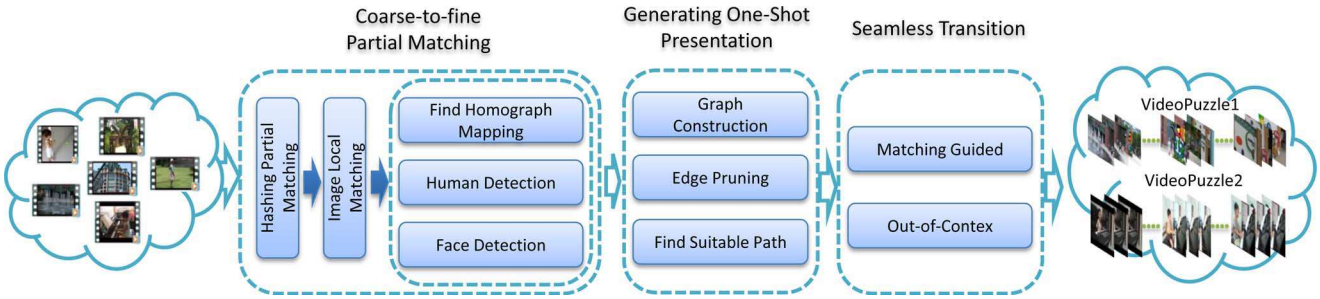


Fig. 2. The illustration of the components of Video Puzzle.

Secondly, we design a flexible scheme to select the optimal video compositions from a constructed video matching graph. The video selection task turns out to find the longest path P in the graph $\hat{G}$ by constructing a video matching graph following three criteria, i.e., continuity, completeness and diversity. This selection scheme can either work fully automatically by creating one-shot videos with globally optimal content consistency or work interactively with users by generating one-shot videos with optional topics (such as the specified key objects or persons). We will introduce the details in Section V.

Finally, we compose the video correspondence pair one by one. We propose a space-temporal morphing-based transition through matched local patterns, i.e., matched local common pattern, matched human or face. The produced transition is more natural than the traditional transitions such as fade-in, fade-out, wipes, and dissolve.[4] Since both image-level and sequence-level matching for video pairs are available, we can accomplish a content-based continuous transition. The proposed content-based transition produces virtually consistent link for the final composition.

The Video Puzzle system, which can automatically generate a virtual one-shot presentation from multiple video clips, provides a novel presentation of video contents and enables users to have a deeper impression of the story from the video collection. We will provide two applications in detail.

IV. Coarse-to-Fine Partial Matching

In this section, we introduce the first component of the system, namely, coarse-to-fine partial matching. The target of this part is to (a) produce video similarity measurement acting as evidence for ensuring the *non-redundant* and *complete* quality of the generated video; (b) fast and accurately locate the sequences in video pairs with start-end content correspondence; and (c) find the keyframe pairs with transition clues in the correspondence sequences. We first use a hashing-based method to quickly obtain the video similarity measurement. Then we try to match two video sub-sequences in order to generate continuous transition. Finally, specific transition clues are obtained for video composition through local common pattern discovery, human appearance modeling and face appearance modeling.

[4]http://en.wikipedia.org/wiki/Shot_transition_detection

A. Hashing-Based Video Pair Selection

In this part, we adopt the recently proposed Partition Min-Hashing (PmH) [24] algorithm to rapidly calculate the frame partial similarity between every pair of videos and the computed frame similarity is accumulated to estimate the video similarity measurement. Then, the video pairs with high similarity are selected as candidate pairs to generate one-shot videos. A graph of video similarity is built based on the results of video pair selection.

In practice, we filter out most video pairs with low similarity and only retain up to four video pairs as the matching candidates for each video. Therefore, the computational cost for the further video matching steps is largely reduced.

1) Min Hashing: Min-hash is a Locality Sensitive Hashing scheme [9] that approximates the similarity between sets. In the min-hash algorithm, a hash function is applied to all visual words in an image without considering their locations, and the visual word with minimum hash value is selected as a global descriptor of the given image. When an image is represented by a set of visual words, the similarity between two images can be defined as the Jaccard similarity between the two corresponding sets of visual words I_1 and I_2, i.e., $sim(I_1, I_2) = I_1 \cap I_2 / I_1 \cup I_2$, which is simply the ratio of the intersection to the union of the two sets. Min-hash is a hash function $h : I \mapsto v$, which maps a set I to a value v. More specifically, a hash function is applied to each visual word in the set I, and the visual word that has minimum hashed value is returned as the min-hash $h(I)$. The computation of the min-hash of a set I involves the hash of every element in the set and the time cost thus scales linearly with the size of the set I. In our case, we are interested in finding images which have similarity greater than a threshold θ. In other words, we would like the probability of collision to be a step function:

$$P(h(I_1) = h(I_2)) = \begin{cases} 1, & \text{if } sim(I_1, I_2) \geq \theta; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

This step function can be approximated by applying k min-hash to a set and concatenating them into a sketch. Then, n sketches can be computed for an image and all of them can be added to the hash table. Under this setting, two images will collide if they share an identical sketch. The probability for two images to collide in the hash table becomes:

$$P(h(I_1) = h(I_2)) = 1 - (1 - sim(I_1, I_2)^k)^n, \quad (2)$$

which approximates the step function in (1). The sharpness of the "step" and the threshold θ can be controlled by varying the sketch size k and the number of sketches n.

2) Partition Min-Hash: However, unlike text documents which are usually represented with bags of words, images are strongly characterized by their 2D structured objects which are often spatially localized in the image. Partition min-Hash (PmH) [9] is proposed as a novel hashing scheme to exploit the locality. In PmH, an image is first divided into partitions. Hashing is then applied independently to the visual words within each partition to compute a min-hash value.

With evenly divided partitions, the duplicate may be split into two or more partitions. To alleviate this, PmH designs partitions to be overlapping and of multi-scale. An image is divided into grids, where the grid elements are the greatest common regions among partitions that cover that region. Min-hash sketches are pre-computed for each grid element g_i. The min-hash sketch for a partition P is then computed by looking up elements $\{g_i\}$ that are associated with that partition P and picking the true min-hash sketch among the pre-computed min-hash sketches on elements:

$$h(P) = min\{h(g_i)|g_i \in P\}. \tag{3}$$

3) Video Similarity Estimation: We first extract the frames $\{V_{i,m}\}$ for each video V_i. The video similarity measurement is then defined by:

$$W_{i,j} = \sum_m \sum_n \frac{\delta(V_{i,m}, V_{j,n})}{|V_i| \cdot |V_j|}, \tag{4}$$

where $\delta(I_1, I_2)$ is defined to be 1 if one sketch of partitions in I_1 collides with other sketches of partitions in I_2 following PmH sketch collision scheme addressed in previous sections, and and $|V_i|$ denotes the frame number of video V_i.

B. Sequence Matching

In this subsection, we aim to accurately match two video subsequences within the selected video pairs in order to generate continuous transition. We propose a method that uses image local matching to get the correspondence of two sub-sequences.

1) Image Local Keypoints Matching: We use SIFT [27] + Color Moments with Difference of Gaussians (DOG) keypoint detector. Existing studies demonstrate that the SIFT descriptors and Color Moments are complementary to each other, one describing the local structure and the other providing higher order information of local differences. We concatenate these two features to describe each local keypoint. To determine the local match, we use the method proposed by [27]. Given two frames (the source image, frame m in video V_i, and the target image, frame n in video V_j), the best candidate match for each keypoint of the source image is found by identifying its nearest neighbor among the keypoints from the target image. The nearest neighbor is defined as the keypoint with the minimum Euclidean distance. Since there will be many keypoints from the source image that do not have any correct match in the target image, such as those that arise from background clutter or are not detected in the target image, it is useful to discard them. An effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. We then get the keypoints matching set KS and their matching scores T which is the similarity measurement of the keypoint matching pair. The image similarity score determined by local matching is defined as:

$$score_{m,n} = \sum_{k \in KS} \frac{T_k}{|KS|^2}, \tag{5}$$

where $|KS|$ denotes the size of the matching set of the frames.

2) Frame Similarity to Sequence Correspondence: To locate the sequence correspondence of two videos, we sample the video frames in a constant rate. Given two frame sequences Seq_i and Seq_j of same length k in two videos, we first calculate the maximum similarity over the frames in Seq_j for each frame in Seq_i. The sequences similarity is then defined as the average of the maximum values, namely:

$$VS_{i,j} = \sum_{m \in Seq_i} \max_{n \in Seq_j} \frac{score_{m,n}}{|k|}. \tag{6}$$

Since the ultimate video clip is expected to contain only one shot, we compose two videos only in their starting or ending part in order to keep the storyline within the clips. Thus the video sequence correspondence is also found within the starting and ending part.

For detail, the video is first partitioned into two parts with equivalent duration. The sequence correspondence $S_{i,j}$ represents a sequence in the second part of the video V_i is matched with a sequence in the first part of the video V_j. Then, the sequence similarity is scaled by a preference factor which is set to 1 when the two sequences are close to the start or end of the videos and gradually turns to 0 when one of the sequences is far from the video border. For each video pair V_i and V_j, we obtain the sequence correspondence $S_{i,j}$ by finding the sequence pair with the largest sequence similarity. The video similarity in the graph is also replaced by this sequence similarity.

This process of searching for sequence correspondence is critical in our system. It determines whether video clips can be composed with the other videos. We will also discuss how to use the video matching score in Section V.

C. Transition Clues for Video Composition

Given two matched sequences, we select the frame pairs $M_{i,j}$ as the transition key frames according to several transition clues.

1) Cross-Frame Common Pattern Discovery: The first transition clue we use is based on image matching. Since image matching often contains a large amount of outliers, we need a robust fitting method to find the common pattern. Specifically, common pattern denotes those matching pairs that share the same or similar homogeneous transformation parameters. RANSAC method is utilized to find the matching transformation parameters. However, practically RANSAC may perform poorly when the ratio of inliers falls below 50%. It means that a large overlap between the pair of images is required for the matching, which is rare case for location representation in a video. Therefore, we need an extra method to determine the true matches within the matched pairs. It is worth noting that [19]

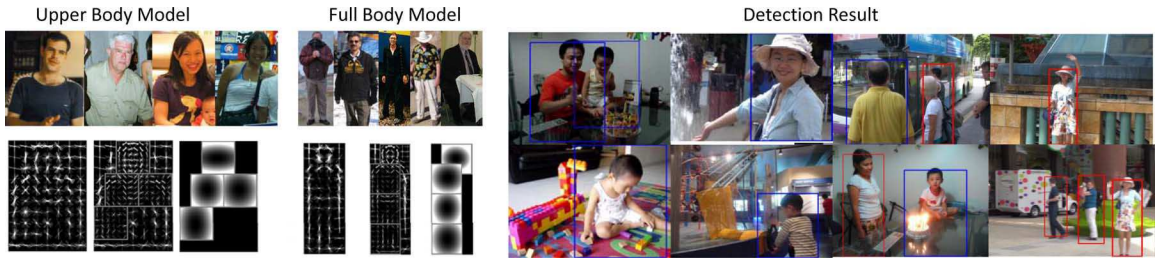


Fig. 3. Matching via human appearance. The figure illustrates several training samples (top-left), human models (bottom-left) and example detection (right) using the part-based human detection model.

incorporated the local matching within a large image collection scenario with RANSAC. For this part, we adopt the "Graph Shift" method [26]. The main idea is to introduce spatial constraint for the matched pair to find a dense common pattern. This algorithm has three advantages over RANSAC: 1) it is robust to outliers; 2) it is able to discover all common visual patterns, no matter the mappings among the common patterns are one-to-one, one-to-many, or many-to-many; and 3) it is computationally efficient.

2) Human Appearance Matching: The frames from two videos are also matched according to the appearance of human contained in the video. Firstly, automatic human body detection is accomplished. We implement the part-based model in [15] learnt with the annotated human images from the PASCAL Visual Object Classes (VOC) Challenge 2010 dataset [14] for human detection. Some examples of the training samples are shown in Fig. 3. The part-based detection model contains two parts, one describing full view (denoted as root model) and the other describing part views (denoted as part models). An illustration of the part-based models is shown in Fig. 3. In our training process, the model is configured to contain 2 root models (for upper body and full body respectively) and 4 part models for each root model.

Fig. 3 demonstrates several exemplary detection results over some frames used in our experiments. The appearances of the detected human bodies are represented in color histogram and matched to find the same person appeared in different frames/ videos.

3) Face Appearance Matching: We also implement the state-of-the-art multi-view face detector [18] and active shape model [10] for face alignment. For each frame, we perform the near-frontal face detector to localize the face area as well as several facial parts, such as eyes, mouth, nose and face contour. A frame with face is assumed to be matched with another frame with face according to the following criteria:

1) Both face areas should be large enough. Small face areas are much less important since video matching and transition on small area frequently lead to unnatural effects. In our implementation, we set the threshold to 3,600 pixels.
2) The faces should belong to the same person. We first perform the face alignment procedure to align the faces and then calculate the Euclidean distance for the feature vectors extracted from each face pair. A threshold is empirically set to remove most mismatched candidates.
3) The two face poses should not vary much. The output of the face detector [18] includes the pose view information. Also, we prefer front face matching than non-front face matching.

Fig. 4. Matching via face appearance. The figure shows the ASM model for face alignment across different video clips.

An example matching is shown in Fig. 4.

Based on the above three transition clues, we locate candidate key frame pairs $M_{i,j}$ from two sequences as follows:

Frame pairs with the same object. For each pair of frames in the sequence correspondence, we perform common pattern discovery, and the frame pair with the maximal pattern support is then chosen as a key frame pair. **Frame pairs with the same person**. If the matched score of persons/faces within a frame pair is greater than a predefined threshold, the frame pair is chosen as a key frame pair.

V. Generation of One-Shot Presentation

Given a video collection that contains N video clips $\{V_i, i = 1, \ldots, N\}$. We construct a matching graph $G = \langle H, E \rangle$ where H_i denotes the i th video, $E_{i,j}$ is the directed weight for the node i and j, which is the maximal sequence matching score $S_{i,j}$ of the video V_i and V_j. Our task here is to find a path P from the directed graph G to connect the short shots into a long-take shot. An intuitive example is illustrated in Fig. 5.

The following criteria are considered to find a path over the graph:

- **Continuity**: Each edge on the path should have a weight greater than a predefined threshold. Otherwise, the edge is removed.

$$E_{i,j} = \begin{cases} E_{i,j}, & \text{if } E_{i,j} \geq \theta, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

- **Completeness**: The overall path should be sufficiently long. To ensure the completeness of the video, large number of combined clips is preferred.
- **Diversity**: The nodes should have large variety. Since the matched clips possibly contain many near-duplicate versions, we need to exclude them to retain the compactness

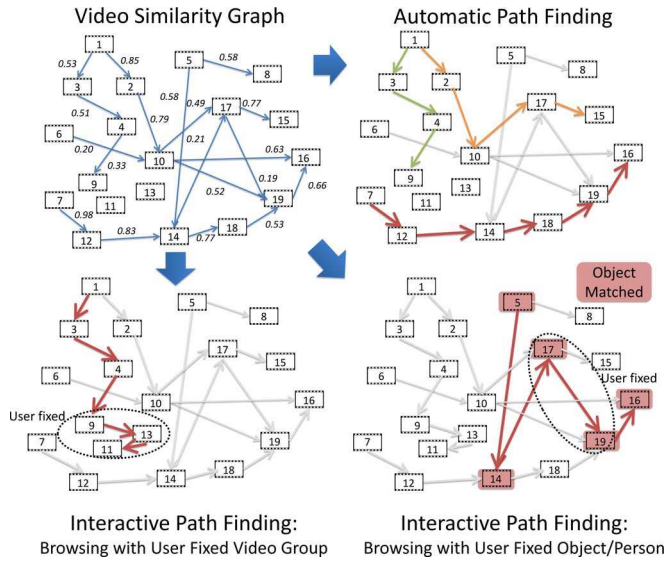


Fig. 5. Graph Construction and Path Finding. The top-left graph is the original video similarity graph for a given video album. The videos are linked with different edge weights. Top-right graph shows the several paths of automatic path finding (indicated by different colors). Bottom-left graph shows the result of interactive path finding with user fixing a small video group (circled in the graph). Bottom-right graph shows the result of interactive path finding with user fixing the matched object/person discovered by the system (circled in the graph).

of the composited video. This step is accomplished by exploring the similarities among videos.

Therefore, the edge of the final graph is weighted by:

$$\hat{E}_{i,j} = \begin{cases} \frac{E_{i,j}}{W_{i,j}}, & \text{if } E_{i,j} \geq \theta, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

The task then turns out to find the longest path P in the graph $\hat{G} = \langle H, \hat{E} \rangle$. The longest path problem can be reduced to the shortest path problem (although the graph may have negative-weight cycles) by exploiting the duality of optimizations (maximizing a positive value equals to minimizing a negative value). If the input graph to the longest path problem is $\hat{G}$, the shortest simple path on the graph $\hat{G}'$, which is exactly the same as G but with inverse edge weights, is the longest simple path on G. However, positive-weight cycles in the original graph G lead to negative-weight cycles in $\hat{G}'$. Finding the shortest simple path on a graph with negative-weight cycle is therefore also NP-complete. If G contains no cycle, then $\hat{G}'$ will have no negative-weight cycle, and any shortest-path finding algorithm can be implemented on $\hat{G}'$ to solve the original problem in polynomial time. Thus, the longest path problem is easy on acyclic graphs. If G is a directed acyclic graph, the longest path problem on G can be solved in linear time using dynamic programming.

A. Edge Pruning

We prune the cycle paths in the graph G to avoid the repeated clips in the composed video. An important and also the most straightforward criterion is time constraint. People are used to watching videos in the order of time, especially for home video browsing. We use the timestamp metadata of the video clips to ensure that the shots maintain the temporal relationships in the composition process. However, we also notice that many video clips lack of such metadata. Therefore, for those video clips, we need to design extra content-based edge pruning method to reduce the cycle graph. Here we use Depth-First-Search [35] to detect all the nodes that have a cycle in the graph. We can locate the edges within the cycle, then the edge with lowest weight will be pruned.

B. Path Finding

1) Automatic Path Finding: The maximal paths can be found automatically. After finding the longest path P over the graph $\hat{G}$, all the edge weight linking to those nodes in the path should be scaled by a factor λ ($\lambda = 0.2$ in our implementation) to reduce the possibility for these nodes to be selected again. We then find the longest path again in the updated graph. This procedure can be iterated until reaching the criterion that the sum of weights in the final path is less than a threshold.

2) Interactive Path Finding: For personal usage, the one-shot technique can help to find and composite consecutive video clips with human interaction. A user may expect a one-shot video that contains a specified key video clip or focuses on a specific object or scene. Our framework is flexible in handling these situations, as illustrated in Fig. 5.

User fixing one video clip V_i: We find the maximal path from other nodes to node V_i and the longest path from node V_i to other nodes. The overall one-shot video is then generated with the combined path. Since the constructed graph is an acyclic graph, it is guaranteed that no node will be selected more than once.

User fixing a group of video clips: User may want to fix several similar video clips into the composition. First, the group of video clips is deemed as a virtual node. All the edges linking to the group clips is linked to this node. The problem then turns out to find the path passing through the node.

User fixing the matched object: We can list many matched objects within the video album, such that user can select a matched object. To find a one-shot video that contains this matched object, we first locate the two video V_i and V_j that contain the selected object, and the problem then turns to finding the longest path that ends at the node V_i and starts at the node V_j.

VI. Seamless Video Composition

Here we introduce how to compose the selected video clips into one-shot video and the key problem is to smooth the visual discontinuities at the transitions. For each two best matched frames, all the matches are local, such as common patterns, human bodies, and faces. Since directly stitching the two videos based on these two frames may lead to abrupt change, we need to consider adding natural transition, which act as the link between the two consecutive videos, in the final virtual long-take video.

In video animation, transition is often accomplished by image morphing [34], [42]. The goal of morphing is to generate the in-between geometry which smoothly transforms the source shape into the target shape with interpolated texture smoothing. Morphing can produce appealing result for matched objects, but it may also cause the *ghost* phenomenon in transition for unmatched parts. This problem is even worse for our task since the video transition is based on partial matching. To tackle the

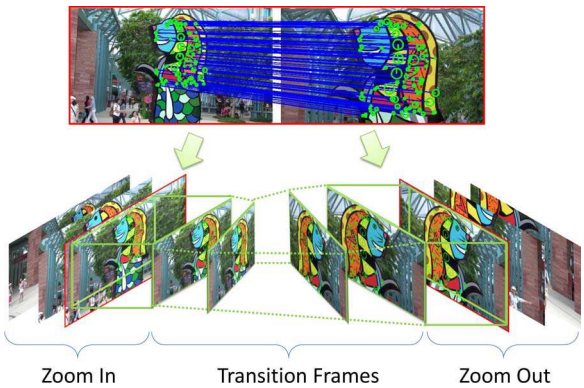


Fig. 6. An illustration of the proposed matching-based transition.

problem, we use the following procedures to generate the more natural transition between videos:

Finding the minimum matched area: The transition between two matched objects often needs to be smooth and continuous. Thus, instead of simply generating a transition between the frames I_i and I_j in the video V_i and V_j, we generate a transition between the largest matched sub-windows M_i and M_j. We select the matched areas by taking three factors in consideration to guarantee the smoothness: (1) keeping the width-height ratio of sub window; (2) finding the minimum matched area covering most of local matched points; and (3) the offset between the centers of the local matched points within M_i and M_j should be minimized.

Focusing on the matched object: After locating the sub-windows M_i and M_j, we find k frames before the frame I_i in the video V_i and produce Zoom-In effect in the sequence. Here k is determined by the area ratio of M_i and I_i. Similarly, Zoom-Out effect is produced on video V_j.

Local alignment and local texture shape mapping: We then process the morphing between M_i and M_j to generate the intermediate frames. Given the matching point set, we first generate the Delaunary triangulation sets P_i and P_j that contain N_p elements. The texture within the triangulation is linearly interpolated. The intermediate triangle $P(t)$ is computed as:

$$P^m(t) = \{tP_i^m + (1-t)P_j^m\}, t \in [0,1], m = 1, \ldots, N_p. \quad (9)$$

Feathering on unmatched area: The transition M_i and M_j may still have $ghost$ effect for the unmatched area. To address this issue, we adopt the feathering approach commonly used for image mosaics. That is, we weight the pixels in each frame proportionally to edge and their distance to the matching points center [33].

Fig. 6 demonstrates illustrative examples and Section VII-C1 will introduce several real examples. Finally, we would like to mention that there might be cases that no matching between two videos could be found. This can be regarded as a "out-of-context" composition. We can use direct composition or adopt other transition methods, such as fade-in and fade-out. An alternative approach is to use the Picture-in-Picture[5] technique. The two videos can be connected through finding a flat area in the end part of the former video and then the latter video can be embedded into this area with a virtual TV frame. The transition can be made through zoom-into the TV frame.

[5]http://en.wikipedia.org/wiki/Picture-in-picture

TABLE I
COMPUTATION COST OF GRAPH CONSTRUCTION

Album	TotalFrames	FeaExtract	PmH [24]	SimilarityCompute
$VA1$	11k	1412s	21.6s	5s
$VA2$	48k	5180s	86.2s	20.3s
$VA3$	23k	2014s	40.3s	11.2s

VII. EXPERIMENTS

A. Dataset Preparation

Several experiments were performed to verify the effectiveness of the proposed "Video Puzzle" framework. Three video albums are prepared for the experiments. We denote them as $VA1$, $VA2$ and $VA3$, where $VA1$ and $VA2$ are typical home video albums and the videos in $VA3$ are collected from Youtube.com with some keywords of famous landmarks. The $VA1$ set contains 68 video clips. They are captured in a trip and the locations vary widely, including beach, landscape, and woods road. The $VA2$ set contains 186 video clips that record the birthday parties of a child, piano practice scenes, etc. The $VA3$ set contains 33 video clips of the famous landmarks "Roman Colosseum" and "Eiffel Tower" including several noise videos from direct YouTube search by keywords. As mentioned in Section III, we segment the video clips into shots before performing our approach.

B. Implementation Details

For each video album, we first construct a bag-of-words (BoW) model using SIFT features for the PmH hashing [24]. The number of features per image ranges from 200 to 1000, and we have quantized them using a visual word vocabulary with one million visual words. It takes about 100 ms per image to extract the features. The min-hash contains 2 sketches of size 2. Each image is divided into about 100 partitions with 50% overlap as recommended in [24]. We uniformly sample 1/5 of the frames to accelerate the video similarity measurement. The frame similarities are accumulated to form video similarity. We then set a threshold to make the similarity graph sparse. For each connected video pairs, local matching-based sequence matching is performed. Finally, exact matching frame pairs are located.

Computation Cost: Table I lists the computation cost for each steps of graph construction including the feature extraction, PmH Hashing and computing the video similarities for each Video Album. As can be seen that the main computation cost has been the feature extraction part which is inevitable for content understanding. Meanwhile, the hashing schemes provide us a great efficiency in measure the similarity of two images. The path finding is almost instant given that the complexity is $O(V^2E)$ where V, E are the number of nodes and edges of the graph respectively. Given one path with N video clips, $N-1$ composition is needed. For each composition, the time cost is about 3.5 seconds including all the effects. All the experiments are conducted on an Intel 3.0 GHZ PC with 16 GB memory.

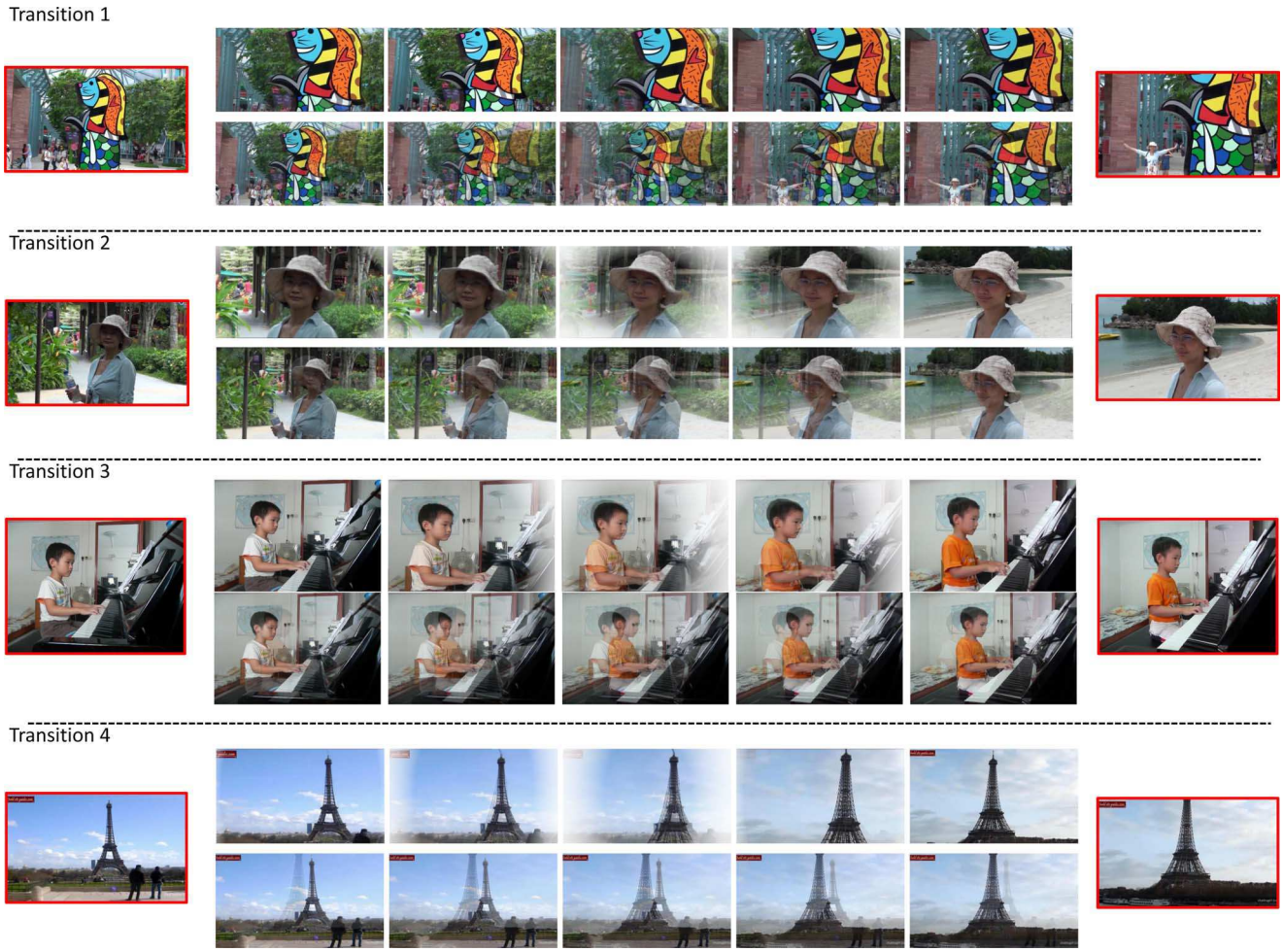


Fig. 7. Comparison of the effects of transitions. For each transition, the matched start and end frame are marked in red. The first row of each transition (we have selected 5 frames) is obtained by the proposed scheme. The second row of each transition is obtained through the widely-used transition effect, i.e., fade-in/fade-out. The transition 1 is generated through matched key points without feathering effect. The transitions 2 to 4 are obtained using the proposed method with different transition clues.

C. One-Shot Video Generation

In this section, we give some generated one-shot examples from the video album $VA1$, $VA2$ and $VA3$.[6]

1) Transition Effect Evaluation: We first check the effect of the generated transitions. We compare the proposed transition method with the widely-used method, i.e., fade-in and fade out. Given two corresponding frames, we compare the generated results. The results are shown in Fig. 7. It can be observed that:

- The transition generated by the proposed method looks much more smooth and seamless than the fade-in and fade-out method. The main objects are still clear enough to identify when using our method in the transitions 1 to 4. The fade-in and fade-out method brings a lot of blurring and ghost. The main advantage of proposed method is that we first automatically focus to the main object within the frames in order to avoid the displacement. The morphing between the matched objects is visually pleasing.
- The feathering method can reduce the ghosting effect. The result of transition 1 is without feathering effect. We can still get some blurring and ghost effect, especially on the border area where there exists most disagreement among the two frames.

2) Family Video Browsing: By taking the $VA1$ as an example, we show the video matching graph after edge pruning in Fig. 8. Each node represents a video in $VA1$. The edge linking two nodes indicates a match between them. The matched region or object is shown upon the edge with a confidence identifying the sequence similarity.

The matching graph brings a new tool for video album browsing. Unlike the traditional video summarization methods which try to extract key content of video album and present in an concise manner, the matching graph has several characteristics: (1) It presents the video album in a wide-range manner and gives the user the direct impression over the large numbers of clips. (2) More importantly, the matching graph gives the correspondence information of two videos. The linking edge introduces the browser to explore the whole set with continuous

[6]As the video sizes are too large as supplementary material, all the generated "Video Puzzles" are available at http://www.youtube.com/theoneshotvideo.

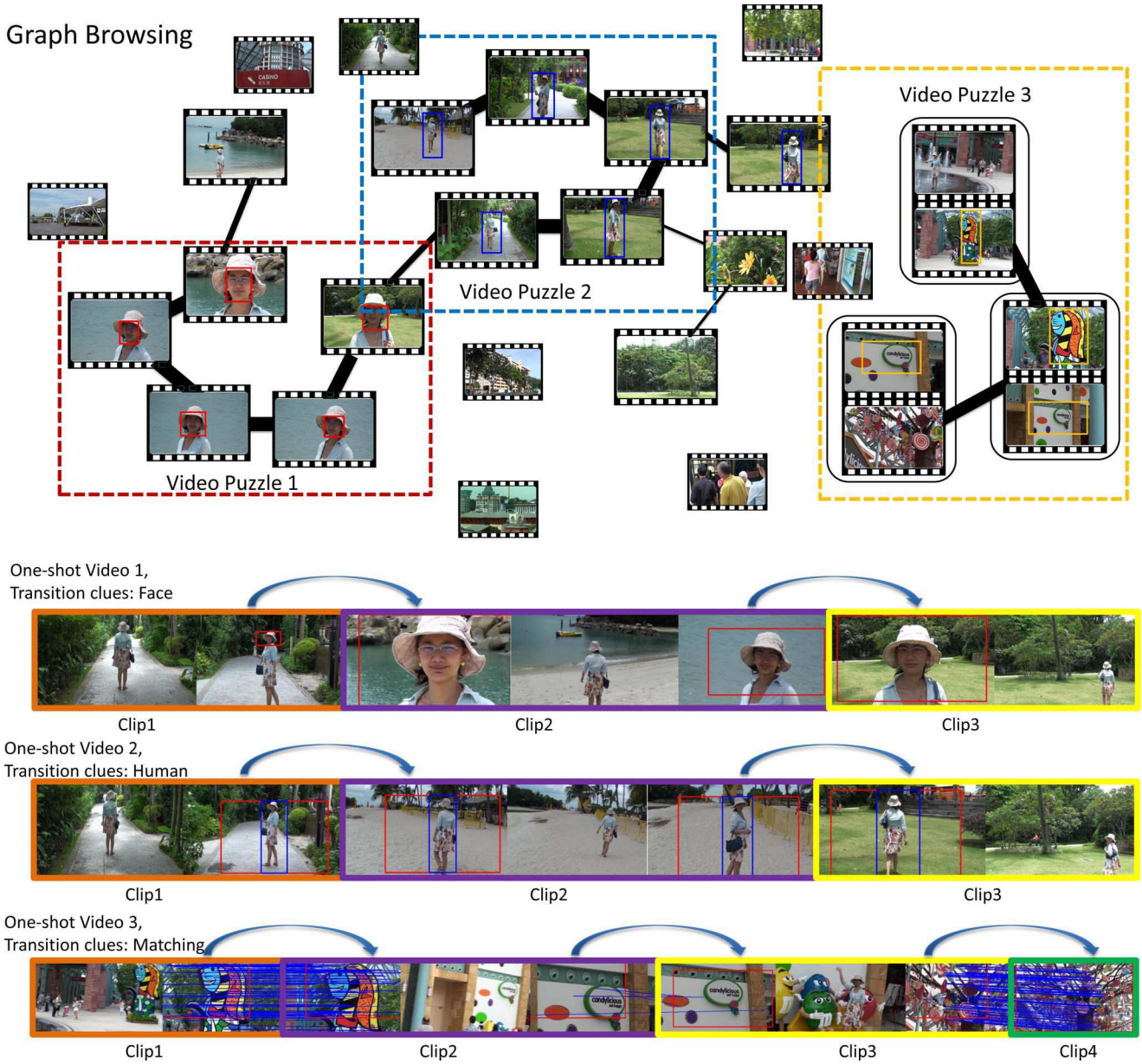


Fig. 8. Example of graph browsing and the one-shot videos in $VA1$. The graph browsing model enable viewers to explore the video clip album efficiently even containing a lot of out-of-context clips. The video-video connection are established through different clues, such as faces, human appearances and matching. The compact sets of clips that can be contained in a one-shot video are automatically popped-up through the proposed scheme. Here we illustrate three compact sets in $VA1$. Two or three frames from the starting, middle and ending part of each video clip are shown (each bold color rectangle box denotes one video clip, different colors represent different clips). The red rectangles and blue lines demonstrate the matched person, face and object key points in consecutive video clips. For more informative results, please see the video in [1].

content rather than one by one. (3) It also highlights the "key" clip in the video album (connected with thick lines in Fig. 8). The "key" clip has the most number of edges connecting from and to it. (4) It provides an interactive interface for user. When user clicks certain video clips, the system can display the linked compact set of video clips that contains this video clip. If the user prefers certain person on the graph edge, the system can generate a consistent one-shot video presentation focusing on the person. The interaction is flexible and has many potential variation.

It can be shown that the video album $VA1$ contains several compact sets, which can generate one-shot videos, and some isolated nodes. Three examples of the generated one-shot videos are demonstrated in Fig. 8, which respectively shows the human appearance, face appearance and object key point induced matching clues. It is worth noting that these One-shot Videos are obtained by automatic path finding introduced in Section V-B.

For the video album $VA2$, timestamps are available, and thus we first prune the graph edges with the timestamp information. Multiple one-shot videos are then generated by the method introduced in Section V-B. It is worth noting that the One-shot Video 1 is obtained by fixing the matched object introduced in Section V-B2. And One-shot Video 2 is obtained by automatic

Fig. 9. Two examples of the one-shot videos generated from $VA2$. Two or three frames from the starting, middle and ending part of each video clip are shown (each bold color rectangle box denotes one video clip, different colors represent different clips). The red rectangles and blue lines demonstrate the matched person, face and object key points in consecutive video clips. For more informative results, please see the video in [1].

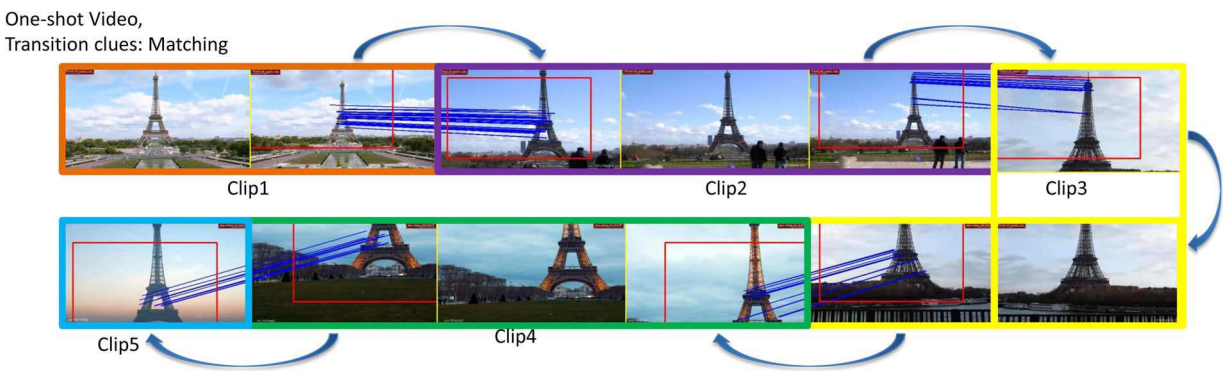


Fig. 10. An example of the one-shot videos generated from $VA3$. Two or three frames from the starting, middle and ending part of each video clip are shown (each bold color rectangle box denotes one video clip, different colors represent different clips). The red rectangles and blue lines demonstrate the matched key points in consecutive video clips. For more informative results, please see the video in [1].

path finding. Some interesting observation have been found in Fig. 9:

- There exist a lot of human faces within the data, such as father's, mother's and son's. Thus serious restricting the face appearance matching leads to the face identification. Since we have implemented the face alignment, the transition on the face is quite smooth.
- Although we expect the "Piano" scene can be found through matching the piano. The SIFT + ColorMoment never works on the "plain" piano, but the scenes are matched upon the global map hanging on the wall. It leads us to further explore the function of other local descriptors.

3) Landmark Videos From Websites: We also evaluate our method on $VA3$, which contains many online video clips about "Roman Colosseum" and "Eiffel Tower". These two scenes are mutually acting as noise. As shown in Fig. 10, we output the longest path which is a presentation about "Eiffel Tower", composed with 5 shots. The match clue used here is induced by common pattern discovery. Usually, for each frame, about 1000 SIFT features can be extracted. However, the images contain a lot of outliers. The traditional RANSAC method fails in this case while our method still can find the common pattern. Liu *et al.* have explained in [26] that it is mainly because of the large variation of the scene and low video quality.

D. User Study

To subjectively evaluate the results, we compare the following methods: (1) Our "Video Puzzle" with content-based transition (denoted as "VP"); (2) The video produced by direct compositing video clips discovered by the "Video Puzzle" scheme (denoted as "DC"); (3) The automatically edited videos with the videos produced by connecting randomly selected video clips (denoted as "RS"). (4) The video summarization method proposed in [28] (denoted as "VS") groups all video shots into scenes and then generates a skimming for each scene by a graph-based mining method. Then, the skimmings are concatenated into a skimming of the whole video set.

The three sets of videos are used to produce the baselines and the "Video Puzzle". Totally, we extract 8 video puzzles that are composed by 35 video clips. Accordingly, we also composite 8 DC videos with the same clips. Another 8 RS videos are randomly generated within each dataset. 20 evaluators were invited to participate in the user study.

We adopt three evaluation criteria: (1) Enjoyability: it measures the extend to which users feel that the video is enjoyable. (2) Naturalness: it measures whether the users feel the visual appearance of the generated videos are natural. (3) Informativeness: it measures the comprehensiveness of the content of the generated videos.

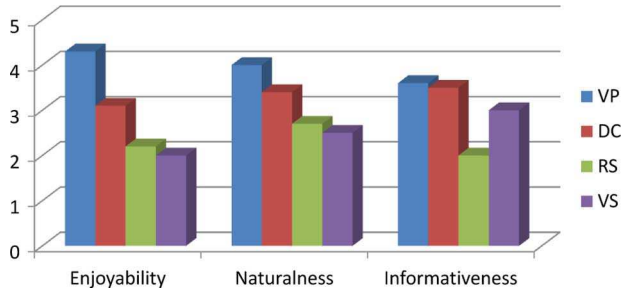


Fig. 11. Diagram of user study results in terms of Enjoyability, Naturalness and Informativeness. The scores are ranged within 1 to 5. The higher the better.

For each criterion, all users are required to provide a score from 1 to 5 for every method (the higher the better). The average scores are shown in Fig. 11. The results show that Video Puzzle has a higher satisfaction than random results and the direct composition. The main reason for this evaluation result is that random selection does not select the important and informative clips. Both VP and DC gets higher scores on informativeness showing the matched video clips explores the main content of video album. VP also gets highest score on enjoyability since it provides novel continuous transition for users. The VS gets low enjoyability and naturalness scores because directly concatenating the shots make the skimming non-smooth. But it is much more informative than the RS method that connects randomly selected clips.

The users are then required to given a final comparison between VP, i.e., the proposed approach, and each of the other approaches (DC, RS and VS) by considering multiple criteria. The users are asked to give the comparison results using $>$, $\gg$, $=$ which mean "better", "much better", and "comparable. To quantify the results, we convert the results into ratings.We assign a score of 1 to the worst scheme, and the other schemes are assigned a score of 2, 3, or 1 if it is better than, much better than, or comparable to this one, respectively. Thus, for each comparison, there are 20 ratings.

Since there will be disagreements among the evaluators, we perform a two-way analysis of variance (ANOVA) test [21], [39] to statistically analyze the comparison. It partitions the observed rating scores into components corresponding to different explanatory factors, and it is able to test the significance levels of the rating differences with respect to the factors of ranking scheme and user. The results are shown in Table II. The results show that Video Puzzle has a higher satisfaction than random results and the direct composition in terms of mean scores. The p-values show that the difference of the different method is significant and the difference of users is insignificant, which is reasonable for that VP can select both important and informative clips from the whole albums.

TABLE II
TWO-WAY ANOVA TEST RESULTS: THE LEFT SIDE OF EACH COLUMN ILLUSTRATES THE MEAN AND STANDARD DEVIATION VALUES OF THE RATING SCORES CONVERTED FROM THE USER STUDY ON THE SATISFACTION COMPARISON OF VP VS. DC, RS AND VS. THE RIGHT SIDE ILLUSTRATES THE ANOVA TEST RESULTS IN TERMS OF F-STATISTIC AND P-VALUE. THE P-VALUES SHOW THAT THE DIFFERENCE OF THE DIFFERENT METHOD IS SIGNIFICANT AND THE DIFFERENCE OF USERS IS INSIGNIFICANT

			method factors		user factor	
	VP	DC/RS/VS	F-stat	p-val	F-stat	p-val
VP vs DC	**2.20±0.36**	1.20±0.31	15.83	8.0×10^{-4}	0.20	0.9995
VP vs RS	**2.60±0.50**	1.05±0.22	83.37	1.0×10^{-9}	0.42	0.9999
VP vs VS	**2.50±0.61**	1.03±0.22	72.96	6.0×10^{-8}	0.45	0.9541

E. On the Robustness of Our Approach

In order to test the robustness of our approach, we perform another test on the three albums. For each album, we randomly selected 20 video clips from the other two albums and added them to the target album as noisy data. We then generated the one-shot videos again. Interestingly, we find that the generated one-shot videos is exactly the same with our previous results. It really shows that our approach can well handle the noisy data. The reasons behind this phenomena are as follows.

1) The video album itself contains noise. As introduced before, VA1 and VA2 are collected from home videos which have large variance. VA3 is collected from youtube.com using keyword search which typically contains a lot of unrelated results.
2) Our coarse-to-find video matching procedure can well handle the false matching problem. The coarse matching prunes the false matching from the global view and the multi-cue matching enables accurate local matching and smooth transition. Although there still exists false matching, these false matching cases often scatter and will not form a global optima during path finding and most of them can thus be excluded. The final one-shot video consisting of several clips is obtained via finding the most confident path. Therefore, the robustness of the system is acceptable although there exist several mismatching cases.

F. Limitation and Future Work

Overall, the Video Puzzle scheme performs well on these three albums. It can automatically discover the consistent topics within personal albums or online landmark albums. It generates consistent video composition based on the semantic matching. However, it also has several limitations: (1) The proposed method only works when the number of videos is large enough and there contains certain consistent topics. (2) The transition clues used in this scheme, i.e., the common keypoints pattern and face/human matching, may produce false alarm. Although the overall path finding optimization tries to find the global optimal solution to avoid the local mismatching, the false matching may still appear in the local path.

In the future, we aim to (1) speed up the feature extraction step to further reduce the time cost and make it a practical system, and (2) visualize the overall video graph with a hierarchical graph structure so that the browsing of the graph can be more efficient.

VIII. CONCLUSIONS

In this paper, we proposed "Video Puzzle", an integrated system for both video summarization, browsing and presentation, based on large amount of personal and web video clips. This system automatically collects content-consistent video clips and generates an one-shot presentation using them. It can facilitate family album management and web video categorization. We demonstrated two example applications using "Video

Puzzle" and the results show that it has great potential to be used in future video management systems.

REFERENCES

[1] The One-Shot Videos. [Online]. Available: http://www.youtube.com/theoneshotvideo.

[2] G. Ahanger, "Automatic composition techniques for video production," *IEEE Trans. Knowl. Data Eng.*, vol. 10, no. 6, pp. 967–987, Nov. 1998.

[3] C. Barnes, D. Goldman, E. Shechtman, and A. Finkelstein, "Video tapestries with continuous temporal zoom," in *Proc. SIGGRAPH*, 2010.

[4] E. Bennett, "Computational time-lapse video," *ACM Trans. Graph.*, vol. 26, no. 102, Jul. 2007.

[5] K. S. Bhat, S. M. Seitz, J. K. Hodgins, P. K. Khosla, K. S. Bhat, S. M. Seitz, J. K. Hodgins, and P. K. Khosla, "Flow-based video synthesis and editing," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 360–363, Aug. 2004.

[6] J. Calic, D. Gibson, and N. Campbell, "Efficient layout of comic-like video summaries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 7, pp. 931–936, Jul. 2007.

[7] Y. Caspi, A. Axelrod, Y. Matsushita, and A. Gamliel, "Dynamic stills and clip trailers," *Visual Comput.*, vol. 22, no. 9, pp. 642–652, Sep. 2006.

[8] P. Chiu, A. Girgensohn, and Q. Liu, "Stained-glass visualization for highly condensed video summaries," in *Proc. ICME*, 2004.

[9] O. C. Philbin, "Near duplicate image detection: min-hash and tf-idf weighting," in *Proc. BMVC*, 2008.

[10] T. Cootes, C. Taylor, and D. Cooper, "Active shape models-their training and application," *Comput. Vision Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[11] C. Correa, "Dynamic video narratives," *ACM Trans. Graph.*, vol. 29, no. 4, Jul. 2010.

[12] C. C. Nikolaidis, "Video shot detection and condensed representation. A review," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 28–37, Mar. 2006.

[13] J. E. Cutting, "Representing motion in a static image: Constraints and parallels in art, science, and popular culture," *Perception*, 2002.

[14] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303–338, 2010.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[16] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. J. Guibas, "Image webs: Computing and exploiting connectivity in image collections," in *Proc. CVPR*, 2010.

[17] X. S. Hua, L. Lu, and H. J. Zhang, "Optimization-based automated home video editing system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 572–583, May 2004.

[18] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 671–686, Apr. 2007.

[19] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. Seitz, "Exploring photobios," in *Proc. SIGGRAPH*, 2011.

[20] B. Kim and I. Essa, "Video-based nonphotorealistic and expressive illustration of motion," in *Proc. CGI*, 2005.

[21] B. King, *Statistical Reasoning in Psychology and Education*. New York, NY, USA: Wiley, 2003.

[22] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *Proc. SIGGRAPH*, 2008.

[23] V. Kwatra, A. Schodl, I. Essa, and G. T. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, Jul. 2003.

[24] D. Lee and Q. Ke, "Partition min-hash for partial duplicate image discovery," in *Proc. ECCV*, 2010.

[25] J. Lee and J. Oh, "Scenario based dynamic video abstractions using graph matching," in *Proc. ACM MM*, 2005.

[26] H. Liu and S. Yan, "Robust graph mode seeking by graph shift," in *Proc. ICML*, 2010.

[27] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] S. Lu, I. King, and M. R. Lyu, "Video summarization by video structure analysis and graph optimization," in *Proc. ICME*, 2004.

[29] T. Mei, B. Yang, S.-Q. Yang, and X.-S. Hua, "Video collage: Presenting a video sequence using a single image," *Visual Comput.*, vol. 25, no. 1, pp. 39–51, Dec. 2008.

[30] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg, "Dynamosaics: Video mosaics with non-chronological time," in *Proc. CVPR*, 2005.

[31] J. Scharcanski and W. Gaviao, "Hierarchical summarization of diagnostic hysteroscopy videos," in *Proc. ICIP*, 2006.

[32] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, "Video textures," in *Proc. SIGGRAPH*, 2000.

[33] R. Szeliski, "Creating full view panoramic image mosaics and environment maps," in *Proc. SIGGRAPH*, 1997.

[34] A. Tal, "Image morphing with feature preserving texture," *Comput Graph. Forum*, 1999.

[35] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, 1972.

[36] S. Uchihashi, J. Foote, and A. Girgensohn, "Video manga," in *Proc. ACM MM*, 1999.

[37] J. Vendrig and M. Worring, "Systematic evaluation of logical story unit segmentation," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 492–499, Dec. 2002.

[38] J. Wang, C. Xu, E. Chng, and H. Lu, "Automatic composition of broadcast sports video," *Multimedia Syst.*, vol. 14, no. 4, pp. 179–193, 2008.

[39] M. Wang, K. Yang, X.-S. Hua, and H.-J. Zhang, "Towards a relevant and diverse search of social images," *IEEE Trans. Multimedia*, vol. 12, no. 8, pp. 829–842, Dec. 2010.

[40] T. Wang, J. Collomosse, R. Hu, D. Slatter, D. Greig, and P. Cheatle, "Stylized ambient displays of digital media collections," *Comput. Graph.*, vol. 35, no. 1, pp. 54–66, 2011.

[41] T. Wang, A. Mansfield, R. Hu, and J. Collomosse, "An evolutionary approach to automatic video editing," in *Proc. CVMP*, 2009.

[42] G. Wolberg, "Image morphing: A survey," *Visual Comput.*, vol. 14, no. 8/9, pp. 360–372, 1998.

[43] W.-Q. Yan, M. S. Kankanhalli, and J. Wang, "Analogies based video editing," *Multimedia Syst.*, vol. 11, no. 1, pp. 3–18, 2005.

[44] G. Zhang, Z. Dong, J. Jia, and L. Wan, "Refilming with depth-inferred videos," *IEEE Trans. Visual. Comput. Graph.*, vol. 15, no. 5, pp. 828–840, 2009.

[45] J. Kobayashi, C. Bi, and S. Takahashi, "Sophisticated construction and search of 2D motion graphs for synthesizing videos," in *Proc. PSIVT*, 2010.

[46] I. K. Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz, "Exploring photobios," *ACM Trans. Graph.*, vol. 30, no. 4, Jul. 2011.

[47] J. S. Boreczky, A. Girgensohn, G. Golovchinsky, and S. Uchihashi, "An interactive comic book presentation for exploring video," in *Proc. CHI*, 2000.

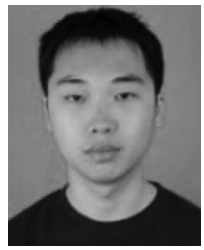
Qiang Chen is a Ph.D. candidate of the Electrical and Computer Engineering Department of National University of Singapore. He received his B.E degree and M.S. from the Department of Automation, University of Science and Technology of China (USTC) in 2006 and Department of Automation, Shanghai Jiaotong University in 2009. His research interests include computer vision and pattern recognition. He received the best student paper awards PREMIA'12 and the winner prizes of the classification task in both PASCAL VOC'10 and PASCAL VOC'11, the honorable mention prize of the detection task in PASCAL VOC'10.

Meng Wang is a professor in the Hefei University of Technology, China. He received the B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, respectively. He previously worked as an associate researcher at Microsoft Research Asia, and then a core member in a startup in Silicon Valley. After that, he worked in the National University of Singapore as a senior research fellow. His current

research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing. He has authored more than 100 book chapters, journal and conference papers in these areas. He received the best paper awards successively in the 17th and 18th ACM International Conference on Multimedia and the best paper award in the 16th International Multimedia Modeling Conference.

Zhongyang Huang received his Master degree in Information Engineering from Nanyang Technological University (Singapore) in 2001 and Bachelor degree in Biomedical Engineering from Shanghai Jiaotong University (China) in 1993 respectively. From 1994 to 1999, he worked as a senior engineer for medical apparatus development in medical image processing area with China-America Joint Venture Kang Ming Biomedical Engineering Ltd. in China. Since 2001 he has been working as a Senior Staff R&D Engineer of Panasonic Singapore Laboratories in Singapore. During this period, he had been actively involved in the standardization activities such as MPEG (MPEG-2/4/7/21), OMA, AVS, ISMA, NFC in the field of Digital Media distribution, communication and management, particularly in Digital Rights Management area. He had made many important contributions to these standardization groups. Recently he started to work in computer vision and multimedia area for photo indexing and management. He received the winner prizes of the classification task in PASCAL VOC'10/VOC'11. Mr. Huang has authored or co-authored 1 book chapter, 20 technical papers and holds over 15 granted patents with numerous others pending in related fields.

Yang Hua received his Bachelor degree in Electrical Engineering and Automation from China University of Mining and Technology and his Master degree in Software Engineering from Peking University in 2005 and 2008 respectively. Since July 2008, he has been working for the Panasonic Singapore Laboratories in Singapore and he is currently a senior R&D engineer there. He was a joint winner of the classification task in PASCAL VOC 2010 and VOC 2011.



Zheng Song is a Ph.D. candidate of the Electrical and Computer Engineering Department of National University of Singapore. He got his B.Sc. degree from EECS Institute of Peking University in 2007. His research interests are in computer vision and intelligent systems.

Shuicheng Yan is currently an Assistant Professor in the Department of Electrical and Computer Engineering at National University of Singapore, and the founding lead of the Learning and Vision Research Group (http://www.lv-nus.org). Dr. Yan's research areas include computer vision, multimedia and machine learning, and he has authored or co-authored nearly 300 technical papers over a wide range of research topics. He is an associate editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (IEEE TCSVT) and *ACM Transactions on Intelligent Systems and Technology* (ACM TIST), and has been serving as the guest editor of the special issues for TMM and CVIU. He received the Best Paper Awards from PCM'11, ACM MM'10, ICME'10 and ICIMCS'09, the winner prizes of the classification task in both PASCAL VOC'10 and PASCAL VOC'11, the honorable mention prize of the detection task in PASCAL VOC'10, 2010 TCSVT Best Associate Editor (BAE) Award, 2010 Young Faculty Research Award, 2011 Singapore Young Scientist Award, 2012 NUS Young Researcher Award, and the co-author of the best student paper awards of PREMIA'09, PREMIA'11 and PREMIA'12.